5

METHOD AND SYSTEM FOR AUDIOVISUAL REMOTE COLLABORATION

Field of the invention

The present invention relates generally to a method and system
10   for collaborating remotely by visualizing, editing and annotating audiovisual
and text content from various locations either simultaneously or consecutively.
The invention provides a simple and cost effective means for creative
developers (filmmakers, video makers, multimedia authors, advertising
creatives, designers, animators, directors, FX supervisors, etc...) and even
15   trainers to collaborate among themselves or with clients and/or trainees, even
when they cannot be together. It even allows for asynchronous collaboration
in those cases when the parties cannot work at the same time, as when they
are in different time zones.

Background of the invention

20

Creative endeavors and the processes leading to them are
being re-shaped by the constant advance of technology. Parties distributed
around the world are now collaborating on projects that once were managed
and developed from a centralized location. In addition to this, the advent of
25   the Internet has resulted in an explosion of new creative outlets and formats,
ranging from the purely artistic to the openly commercial and everything in
between. These new modes of expression and communication sometimes
require collaboration among parties who cannot meet in person, and
telephone calls, faxes and even e-mails are sometimes not enough to convey
30   adequate information to facilitate the joint work of these parties.

Examples of these are:

2

1. Postproduction and special FX houses working with remote clients: this practice is more and more common and, until the present invention, required high end systems linked by dedicated connections. A well-known example of this was when Steven Spielberg had to supervise the special effects for "Jurassic Park 2: The Lost World" while shooting "Schindler's List" in Europe. The FX team was in California and had to provide daily samples of the work for the director to supervise and approve.

2. Advertising agencies working with clients and websites in remote locations: advertising work often requires multiple revisions, either generated by the client or in order to accommodate specifications of the publisher where the ad is poised to run. Prior to the current invention this entailed communications back and forth between the parties that demanded detailed descriptions and multiple versions.

3. Presentations, training materials and help desks: the state of business in this globalized world demands that organizations be able to train and help end-users and employees remotely, as well as to showcase all kinds of audio visual and presentation materials to distant parties.

Summary of the Invention

The present invention solves the problems described above by enabling real-time visualization of content from multiple locations simultaneously and/or consecutively, and by covering such content with a transparent layer that can hold annotations linked to specific key points in the content, hence preserving its integrity.

A way to visualize this is to imagine a book with old-fashioned acetate transparencies covering each page. Each transparent sheet is associated with a given page and can hold annotations, drawings and edits. All without modifying the underlying text.

The present invention achieves similar results by wrapping digital content (video, audio, animations [vector based or otherwise], multimedia content, word processor documents, slide shows, etc...) in a transparent container. In addition to allowing annotations, this wrapper may enable real time visualization from remote locations.

In accordance with a preferred embodiment, users may interact with content and with one another in real time, by using a standard web browser to log onto a web page containing a workspace that holds both the content and the annotations. This web page can be accessed by multiple

5   users simultaneously, either openly or after entering a password, and may allow for multiple administrators (users with full editorial control) or for a single administrator and a number of viewers (users without editorial control who can only see the material).

Users may then navigate through the content via a timeline and

10   playback controls (also inside the workspace), going from key point to key point (or skipping) and making notes and drawing graphics associated with each of them. Key points can be frames, as can key frames in a movie or an animation, slides in a PowerPoint presentation, or pages in a text document. Annotations can be seen in real time by other logged in users or at a later

15   time. The timeline may indicate visually which key points include annotations, in order to simplify later reference.

The presently preferred embodiment of the invention uses Macromedia Flash (currently in its MX version) to create the workspace with the navigational and editorial tools, encase the content and generate and read

20   XML files describing the annotations. In addition to Flash, the preferred embodiment uses the Macromedia Flash Communications Server. The selected content is first covered with a wrapper that includes tools for navigating, editing and annotating the content, as well as a communication toolkit that connects with a server that enables multiple connections.

25   As should be clear to those skilled in the art, the wrapper and the server could be built using alternative technologies, like Java or Delphi.

Brief Description Of The Drawings

The foregoing brief description, as well as further objects, features, and advantages of the present invention will be understood more

30   completely from the following detailed description of a presently preferred, but nonetheless illustrative, embodiment with reference being had to the accompanying drawings, in which:

Figures 1A and 1B, also referred to collectively as Figure 1, constitute a block diagram illustrating the upload process;

Figure 2 is a block diagram illustrating the login and content selection processes;

Figure 3 is a block diagram illustrating the operation of the invention when displaying video content;

Figure 4 is a block diagram illustrating the operation of the invention when displaying HTML content;

Figure 5 is a block diagram illustrating the operation of the invention when displaying Flash content; and

Figure 6 is a screenshot of the preferred interface.

Detailed Description of the Preferred Embodiments

In accordance with the preferred embodiment of the present invention, users connect with a "Kratzer" website operating on a collaboration server by means of a conventional web browser. This website contains the multimedia content on which users are to collaborate, and permits multiple users to interact with the content simultaneously and with each other. Different users may also access the same content at different times.

On the Kratzer website, a user is presented with an interface that permits him to navigate through multimedia, audiovisual content by means of a timeline and playback controls, manipulating various key points and adding notations or graphics. The user is also able to upload, manipulate and edit various types of objects, including video, Flash and HTML objects.

Figure 6 is a screenshot of an interface disclosed for illustrative purposes, where block A indicates the content viewing area. Block B indicates the navigational controls, which includes controls to operate various content, such as a timeline T and playback controls P. Block C indicates the annotation tools, including tools to add various forms of content. Those skilled in the art will be familiar with these tools and their operation. Block D indicates text and video chat tools. All features are included for descriptive purposes. Features can be added and removed.

In operation content, such as Flash objects, video and audio, may be navigated by means of the timeline and other controls in area B.

Annotations, graphics and such are created, manipulated and edited with tools C, and the user may use tools D for text and video chatting to collaborate in real time with other users.

In general, video, Flash and HTML may be uploaded by a user to the Kratzer site and added to a menu on the interface. This process is illustrated in Figure 1. The user may log on to the Kratzer website and manipulate objects available to him. This process and the selection of objects is illustrated is Figure 2. This process determines whether the object is Flash video or HTML and selects an appropriate process to permit user manipulation, editing and saving of the objects. Figures 3, 4 and 5 illustrate the processes for Flash, video and HTML, respectively.

Figure 1 is a functional block diagram of the object upload process; the figure being divided so that blocks on the left describe operations performed on the client side location and the blocks on the right describe operations performed on the server side location.

Operation begins at block #1 and continues on to #2, where the user selects a file (object) to be uploaded. Block #3 determines whether the selected file is video, if the answer is yes the process continues at block #4 where the object is instanced, then on to #5 where the object is checked for size and appropriate extension, and then on to #6, where it is validated. If the object is valid, then the process continues on to # 7 on the server side, otherwise the process returns to block #2 where users may select other objects. At the server side, objects are checked for file type (#7), if the file type is FLV, then the process continues on to #30. If the file type is other than FLV, then it is converted into FLV at block #8 and then the process continues at block #30.

If the answer at block #3 is negative, the process continues on to block #10; where it is determined whether the selected object is Flash object. If the answer is positive, the process continues on to #11 where the Flash object is instanced and then on to #12, where Flash specific checks are performed. At block #13 object validity is checked, if the answer is positive, then the process continues on to #30, otherwise the process returns to block #2, where users may select other objects.

6

If the answer at #10 is negative, the process continues on to block #20; where it is determined whether the selected object is an HTML object. If the answer is positive, the process continues on to #21 where the HTML object is instanced and then on to #22, where HTML specific checks are performed. Then, block #23 checks object validity, if the answer is positive, then the process continues on to #30; otherwise the process returns to block #2 where users may select other objects.

If the result at block #20 is negative, the process returns to block #2 where users may select other objects.

Otherwise all processing continues at block #30 on the server side, where a new directory is created, then on to #31 where the uploaded object is saved in the new directory and on to #32, where the menu of available objects is updated.

The process ends at block #33.

Figure 2 is a functional block diagram of the operation of the "log on" process and the selection of the content and content type being viewed. Again, the figure is divided into left-hand blocks describing operations performed on the server side and the right-hand blocks describing operations performed on the client side.

Operation begins at block # 101, on the client side, with the client opening a standard browser window and requesting the Kratzer Home page. At block #102, the server receives the request and sends HTML code to the user, who inputs a user ID and password (block #103) in order to log on. At block #104, the server looks up the user in a registered user database and confirms validity of the user in block #105. if the user is valid then processing continues on to block #106, otherwise it returns to #103 for new user and password input.

After the user is validated, block #106 finds the content objects available for such user. If there are no objects available to such user, the test at block #107 routes the process on to #108 and communicates this to the user; otherwise the process continues on to #109, where the user is presented with a list of available objects to choose from.

Once the user selects an object to work with (block #109), the selection is passed on to the server, and at block #110 the server determines

7

whether the content to be seen is Flash based. If the answer is positive, then the process for viewing Flash based content is activated at block #111 and the current process ends.

If the answer to #110 is negative, then the process continues on to #112, where the system determines whether the content to be seen is Video based. If the answer is positive, then the process for viewing video based content is activated at block # 113 and the current process ends.

If the answer at #112 is negative, then the process continues on to #114, where the system determines whether the content to be seen is HTML based. If the answer is positive, then the process for viewing HTML based content is activated at block # 115 and the current process ends.

Figure 3 is a Functional block diagram of the operation of the invention when displaying video content, divided so that the left-hand blocks describe operations performed on the server side and the right-hand blocks describe operations performed on the client side.

Operation begins at block # 301, on the client side, with the request for an ASP file, using parameters determined in the process described in Figure 2.

Block # 303 depicts the server receiving the request from block #301 and splitting the process into two parallel and concurrent sub-processes: the first is at #305, where the server obtains all parameters needed to display the selected object. Simultaneously, in the second process, HTML code including tags requesting an SWF file is delivered to the user and is executed at block #307. On execution on the client side, such HTML tags request the Flash based wrapper (SWF) file. At block #309 the server receives the request and delivers the SWF file.

Block #311 shows the client executing the SWF wrapper file using parameters obtained in the aforementioned parallel process that takes place at block #305. Here the wrapper requests the Video object, as seen in # 313, and the associated XML file where the annotations will reside, # 315.

The process continues at #316, where the video object and XML File ar received, and the user views and annotates the content. On completion of work, block #320 provides a choice to save the annotations. If the answer is positive, the process transfers to #322 where the updated XML file with the

annotations is saved to the server and the process ends at #324. Otherwise, the process ends at #324 without saving.

Figure 4 is a divided functional block diagram illustrating the operation of the invention when displaying HTML content, wherein the left-hand blocks describe operations performed on the server side and the right-hand blocks describe operations performed on the client side.

Operation begins at block # 401, on the client side, with the request for an ASP file, using parameters determined in the process described in Figure 2.

Block # 403 depicts the server receiving the request from #401 and splitting the process into two parallel and concurrent sub-processes: one at #405, where the server obtains all parameters needed to display the selected object. Simultaneously in the second process (#407), HTML code including tags requesting an SWF file is delivered to the user and executed at block #407. On execution on the client side, such HTML tags request the Flash based wrapper (SWF) file. At block #409 the server receives the request and delivers the SWF file.

Block #411 shows the client executing the SWF wrapper using parameters obtained in the aforementioned first parallel process that takes place at #405. Here the executing wrapper requests the HTML object, as seen in # 413, and the associated XML file where the annotations will reside, # 415.

The process continues at #416, where the HTML object and XML file are received, and the user views and annotates the content. On completion of work, control transfers to block #420, which offers the choice to save the annotations. If the answer is positive, the process moves to #422 where the updated XML file with the annotations is saved to the server, after which the process ends at #424. Otherwise, the process ends at #424 without saving.

Figure 5 is a Functional block diagram illustrating the operation of the invention when displaying Flash content, where the left-hand blocks describe operations performed on the server side and the right-hand blocks describe operations performed on the client side.

Operation begins at block # 501, on the client side, with the request for an ASP file, using parameters determined in the process described in Figure 2.

Block # 503 depicts the server receiving the request from #501 and splitting the process into two parallel and concurrent sub-processes, the first of which is performed at #505, where the server obtains all parameters needed to display the selected object. Simultaneously, in the second sub-process (#507), HTML code including tags requesting an SWF file is delivered to the user and executed at block #507. On execution on the client side, such HTML tag requests the Flash based wrapper (SWF) file. At block #509 the server receives the request and delivers the SWF file.

Block #511 shows the client executing the SWF wrapper using parameters obtained in the aforementioned parallel process that takes place at #505. Here the wrapper requests the Flash object, as seen in # 513, and the associated XML file where the annotations will reside in # 515.

The process continues at #516, where the object and XML file are received and the user views and annotates the content. On completion of work, block #520 offers the choice to save the annotations. If the answer is positive, process moves to #522 where the updated XML file with the annotations is saved to the server and the process ends at #524. Otherwise, the process ends at #524 without saving.

Alternate embodiment of the present invention

Alternatively, for those cases when real time collaboration is not possible or not necessary, a Flash wrapper similar to the one in the preferred embodiment, with or even without the communications capabilities can be used sequentially.

The way in which this embodiment functions is as follows: the content is first inserted onto the wrapper and compiled into a self contained SWF file. The resulting file includes the content and the navigation, editing and annotation tools (similar to the preferred embodiment). Once the compiled file is ready it can be distributed electronically in a number of ways

(e.g. via Web, email, disk, network) for others to view and work on. The process can be iterative, allowing for multiple rounds of revisions.

Preferred code for the disclosed process is attached as Appendix A.

5   Although a preferred embodiment of the invention has been disclosed for illustrative purposes, those skilled in the art will appreciate that many additions, modifications, or substitutions are possible without departing from the scope and spirit of the invention. For example, in addition to the preferred and described embodiment, those skilled in the art will easily

10   recognize other ways of achieving similar results in a non-concurrent fashion, allowing users to view the content individually and even offline; annotate and edit it; and then send it to other parties to view the annotations and edits; and even add some of their own. This sequential process can obviously support multiple rounds of revisions.

15

APPENDIX A

Code Section

Code included for illustration purposes only.

Language: Macromedia FLASH MX Actionscript

Initialization process and loading of components and tool palettes; loading of content.

```
Scene 1
    actions for frame 6
        _global.KTZ_CLB_connect = function(skl) {
            //skl: el shoshkele a ver
            _global.KTZ_CLB_connection = new NetConnection();
            KTZ_CLB_connection.onStatus = function(info) {
                KTZ_CLB_connectSO();
            };

KTZ_CLB_connection.connect("rtmp://kratzer.unitedsites.com.ar/kratzer/"+skl,
KTZ_USER);
        };
        _global.KTZ_CLB_connectSO = function() {
            _global.KTZ_CLB_SOFrames = SharedObject.getRemote("frames",
KTZ_CLB_connection.uri, true);
            KTZ_CLB_SOFrames.onSync = function(obj) {
                //la primera vez solo me indica quese conecto
                _global.usedFrames = {};
                for (var i in KTZ_CLB_SOFrames.data) {
                    //delete KTZ_CLB_SOFrames.data[i]
                    usedFrames[i.substr(5)] = true;
                }
                KTZ_CLB_SOFrames.onSync = function(obj) {
                    for (var i in obj) {
                        if (obj[i].code == "change" &&
obj[i].name.indexOf("FRAME") == 0) {
                            var n = obj[i].name.substr(5);
                            if (_level3._frameActual == n) {
                                KTZ_CLB_connectToFrame(n, null);
                            }
                            usedFrames[n] = true;

_level3.KTZ__movieController.contenedor["f"+n].gotoAndStop(3);
                        } else if (obj[i].code == "delete" &&
obj[i].name.indexOf("FRAME") == 0) {
                            var n = Number(obj[i].name.substr(5));
                            if (_level3._frameActual == n) {
                                _level3.KTZ__LIMPIA();
                                KTZ_CLB_SOFrameActual.close();
                                delete KTZ_CLB_SOFrameActual;
                            }
                            delete usedFrames[n];
                            if (n%5 == 0) {
                                var ir = 2;
                            } else {
                                var ir = 1;
                            }

_level3.KTZ__movieController.contenedor["f"+n].gotoAndStop(ir);
```

```
                }
              }
            };
          KTZ_CLB_SOFrames.changeFrame = function(frame, usr) {
              if (frame != _level3.KTZ__movieMC._currentframe && usr !=
KTZ_USER && _level3.KTZ__movieController.free != true) {
                  _level3.KTZ__movieMC.gotoAndStop(frame);
                  _level3.KTZ__movieController.numFrame = frame;
                  _level3._frameActual = frame;
                  _level3.KTZ__LIMPIA();
                  //_root.KTZ__MOSTRAR();
                  if (usedFrames[frame] == true) {
                      KTZ_CLB_connectToFrame(_level3._frameActual, null);
                  } else {
                      _level3.processing._visible = 0;
                      KTZ_CLB_SOFrameActual.close();
                      delete KTZ_CLB_SOFrameActual;
                  }
              }
            };
          _global.ContentFrames = KTZ_CLB_SOFrames.data;
          //loadMovieNum("Comunica.swf", 4);
          //_root.ConectaChat();
          _root.ConectaGlobalNotes();
          _root.KTZ__SYSCARGO++;
        };
        KTZ_CLB_SOFrames.connect(KTZ_CLB_connection);
      };
      ConectaGlobalNotes = function () {
          _global.KTZ_CLB_SOGlobalNotes =
SharedObject.getRemote("GlobalNotes", KTZ_CLB_connection.uri, true);
        KTZ_CLB_SOGlobalNotes.onSync = function() {
            // esta es que se conecto
            _root.ConectaLista();
            KTZ_CLB_SOGlobalNotes.onSync = function(obj) {
                for (var i in obj) {
                    if (obj[i].code == "change") {
                        _level3.globalNotes.Objeto.note.text =
KTZ_CLB_SOGlobalNotes.data.KTZobjTEXT;
                    }
                }
            };
        };
        KTZ_CLB_SOGlobalNotes.connect(KTZ_CLB_connection);
      };
      ConectaLista = function () {
          _global.KTZ_CLB_SOLista = SharedObject.getRemote("lista",
KTZ_CLB_connection.uri, false);
        KTZ_CLB_SOLista.onSync = function(obj) {
            for (var i in obj) {
                trace(obj[i].code+" : "+obj[i].name+" : "+obj[i].oldValue);
            }
            for (var i in KTZ_CLB_SOLista.data) {
                trace(i+"[][][]");
            }
            _root.ConectaChat();
            KTZ_CLB_SOLista.onSync = function(obj) {
                for (var i in obj) {
                    trace(obj[i].code+" : "+obj[i].name+" :
"+obj[i].oldValue);
                }
            };
        };
        KTZ_CLB_SOLista.connect(KTZ_CLB_connection);
      };
      ConectaMouse = function (nombre, tipo) {
          //tipo:1 otro usuario, 0 uno mismo
      }
```

```
        ConectaChat = function () {
            _global.KTZ_CLB_SOTemporal = SharedObject.getRemote("Chat",
KTZ_CLB_connection.uri, false);
            // va a controlar las posiciones de mouse, el frame actual, y el
texto de chat
            KTZ_CLB_SOTemporal.onSync = function() {
                // esta es que se conecto
                loadMovieNum("Comunica.swf", 4);
                KTZ_CLB_SOTemporal.onSync = function() {
                };
            };
            KTZ_CLB_SOTemporal.onChat = function(txt, usr) {
                // _global.KTZ_CLB_CHAT += ">"+usr+": "+txt+newline;
                // trace(">>>" + _global.KTZ_CLB_CHAT);
                //
                // if (_global.KTZ_CLB_CHAT.length>10000) {
                // var dif = _global.KTZ_CLB_CHAT.length-100000;
                // _global.KTZ_CLB_CHAT = _global.KTZ_CLB_CHAT.substr(dif);
                // }
                _global.KTZ_APP_CHAT(txt, usr);
            };
            KTZ_CLB_SOTemporal.connect(KTZ_CLB_connection);
        };
        _global.KTZ_CLB_connectToFrame = function(frame, rtn) {
            if (!_level3._noVer) {
                _level3.processing._visible = 1;
                //rtn: es a que funcion llama despues de conectarse, se trata de
un Objeto en donde fn es la funcion y args son los argumentos
                _global.KTZ_CLB_SOFrameActual = SharedObject.getRemote(frame,
KTZ_CLB_connection.uri, true);
                KTZ_CLB_SOFrameActual.onSync = function() {
                    _level3.processing._visible = 0;
                    _level3.KTZ__MOSTRAR();
                    this.rtn.fn.apply(this.rtn.obj, this.rtn.args);
                    KTZ_CLB_SOFrameActual.onSync = function(obj) {
                        for (var i in obj) {
                            if (obj[i].code == "change") {
                                _level3.KTZ__MOSTRAR_UNO(obj[i].name);
                            } else if (obj[i].code == "delete") {
                                _level3.KTZ__LIMPIA_UNO(obj[i].name);
                            }
                        }
                    };
                };
                KTZ_CLB_SOFrameActual.frame = frame;
                KTZ_CLB_SOFrameActual.rtn = rtn;
                KTZ_CLB_SOFrameActual.connect(KTZ_CLB_connection);
            }
        };
    actions for frame 6
        loadVarsObj = new LoadVars();
        KTZ__SAVE = function () {
            _root.tmpXML = new XML();
            _root.tmpXML = _root.object2XML();
            loadVarsObj.texto = escape(_root.tmpXML.toString());
        };
        KTZ__SEND = function () {
            loadVarsObj.send(_root.KTZ__path+"test_kratzer.asp", "_blank",
"POST");
            delete _root.tmpXML;
        };
        KTZ_CHK_LOAD = function () {
            if (_root.KTZ__SYSCARGO == 2) {
                clearInterval(_root.KTZ_SETINTERVAL);
                _root.KTZ__ONLOAD();
            }
        };
```

```
KTZ__LOAD = function () {
    // _root.XML2object(_root.KTZ__xml);
    _root.KTZ_SETINTERVAL = setInterval(KTZ_CHK_LOAD, 100);
};
KTZ__ONLOAD = function () {
    trace("aaa");
    _level3._root.attachPaletas();
    loadMovieNum(_root.KTZ__path+_root.KTZ__skl, 2);
    _root.preload_L2 = setInterval(preload2, 100);
    _root.loading.texto = "Gathering Shoshkele Info";
    _root.loading.barra.L = _level2;
    _level3._visible = 1;
    ob = KTZ_CLB_SOGlobalNotes.data
    _level3._root.mostrarKTZGNotes("Note","__KTZGlobal_Note", 4,
0x000000, 20, 50, 0, 0,0, 700, 550, 85, ob.KTZobjTEXT);
    _level3._root.KTZ__checkCliks.construyendo = 0;
    _level3._root.KTZ__paletaHerra.onChangeNum();
    // elegir la herramienta default
    KTZ_setDisable();
};
KTZ__RELOAD = function () {
    _level3._root.KTZ__reloj.reloj.stop();
    unloadMovieNum(2);
    loadMovieNum(_root.KTZ__path+_root.KTZ__skl, 2);
    _root.reload_L2 = setInterval(reload_2, 100);
};
function reload_2() {
    _level2._visible = 0;
    var amount =
Math.round((_level2.getBytesLoaded()/_level2.getBytesTotal())*100);
    if (amount == 100) {
        clearInterval(_root.reload_L2);
        _level2._visible = 1;
        _level3._root.KTZ__reloj.reloj.play();
    }
}
actions for frame 6
    //#include "parser2.as"
actions for frame 6
    // initSKL_TML es si tiene timeline
    // para saber cual es la movie que hay que mostrar y habilitar
timeline
    _global.initSKL_TML = function(movie) {
        _level3._root.KTZ__MODE = "TML";
        _level3._root.KTZ__movieMC = movie;
        //aca debe llevar al frame actual
        if (KTZ_CLB_SOTemporal.data.frameActual != null &&
KTZ_CLB_SOTemporal.data.frameActual != undefined) {
            movie.gotoAndStop(KTZ_CLB_SOTemporal.data.frameActual);
            _level3.KTZ__movieController.numFrame = frame;
            _level3._frameActual = frame;
            _level3.KTZ__LIMPIA();
            //_root.KTZ__MOSTRAR();
            if (usedFrames[frame] == true) {
                KTZ_CLB_connectToFrame(_level3._frameActual, null);
            } else {
                _level3.processing._visible = 0;
                KTZ_CLB_SOFrameActual.close();
                delete KTZ_CLB_SOFrameActual;
            }
        }
        if (!_level3._root.controllerExists) {
            _level3._root.attachController();
            KTZ_setEnable();
        }
    };
    _global.initSKL_KEY = function(key) {
        _global.keyActive = key;
```

```
                _level3._root.KTZ__MODE = "KEY";
                if (_level3._root.controllerExists) {

_level3._root.KTZ__paletaHerra.openPal(_level3._root.KTZ__movieController);
                    _level3._root.KTZ__movieController.editEnable(true);
                } else {
                    _level3._root.attachController();
                    _level3._root.KTZ__movieController.inicio();
                }
        };
        _global.updateSKL_KEY_TIMER = function() {
            _level3._root.KTZ__movieController.updateTimer(keyActive.timer);
        };
        _global.endSKL_KEY = function() {
            _level3._root.KTZ__movieController.editEnable(false);
            KTZ_setDisable();
        };
        _global.KTZ_setEnable = function() {
            _global.paletteMSG.broadcastMessage("enableTools");
            _level3._root.itemsMSG.broadcastMessage("enableTools");
        };
        _global.KTZ_setDisable = function() {
            _global.paletteMSG.broadcastMessage("disableTools");
            _level3._root.itemsMSG.broadcastMessage("disableTools");
        };
    actions for frame 6
        Stage.scaleMode = "noScale";
        // sacar todo esto cuando cargue desde un asp!!! --------------------
        /*
        _root.KTZ__path = _url.substring(0, _url.lastIndexOf("/") + 1);
        _root.KTZ__skl = "MOCKUPS/skl_prueba_KEY.swf";
        _root.KTZ__fondo = "FONDOS/back_imdb.JPG";
        _root.KTZ__xml = "baseXML2.xml";
        */
        // ----------------------------------------------------------------

        //if (typeof(MMSave) == "function") {
        _root.KTZ__path = _url.substring(0, _url.lastIndexOf("/") + 1);
        _root.KTZ__skl = "MOCKUPS/video.swf";
        _root.KTZ__fondo = mull//"FONDOS/back_imdb.JPG";
        _root.KTZ__xml = "baseXML2.xml";
        //}
        KTZ_CLB_connect(_root.KTZ__skl)
        _global.paletteMSG = new Object();
        ASBroadcaster.initialize(_global.paletteMSG);
        // orden de carga: interfase, shoshkele, fdo.
        loadMovieNum(_root.KTZ__path + "INTERFACE/interfase_key.swf",3);
        _root.preload_L3 = setInterval(preload3, 100);
        _root.loading.texto = "Building Interfase";
        _root.loading.barra.L = _level3;
        _root.loading.barra.onEnterFrame = function() {
            var amount = Math.round((this.L.getBytesLoaded() /
this.L.getBytesTotal()) * 100);
            this._width = amount;
        };
        function preload1(){
            _level1._visible = 0;
            _root.loading.barra.L = _level1;
            var amount = Math.round((_level1.getBytesLoaded() /
_level1.getBytesTotal()) * 100);
            if(amount == 100 || _root.skipFondo){
                clearInterval(_root.preload_L1);
                _level1._visible = 1;
                delete _root.loading.barra.onEnterFrame;
                _root.loading.texto = "";
                _root.loading._visible = 0;
            }
        }
```

```
        function preload2(){
            _level2._visible = 0;
            _root.loading.barra.L = _level2;
            var amount = Math.round((_level2.getBytesLoaded() /
_level2.getBytesTotal()) * 100);
            if(amount == 100){
                clearInterval(preload_L2);
                _level2._visible = 1;
                _level3._root.KTZ__reloj._visible = true;
                ob = ContentFrames["globalObjects"]["__ELEM__KTZGlobal_level2"];
                _level3.setKTZGLevel2(ob.KTZid, ob.KTZcontenedorX,
ob.KTZcontenedorY, ob.KTZobjSCALE);
                if (_root.KTZ__fondo == undefined || _root.KTZ__fondo == "") {
                    _root.skipFondo = true;
                    preload_L1 = setInterval(preload1, 50);
                } else {
                    _root.skipFondo = false;
                    loadMovieNum(_root.KTZ__path + _root.KTZ__fondo,1);
                    preload_L1 = setInterval(preload1, 50);
                    _root.loading.texto = "Building Background";
                    _root.loading.barra.L = _level1;
                }
                initSKL_TML(_level2)
            }
        }
        function preload3(){
            _level3._visible = 0;
            _root.loading.barra.L = _level3;
            var amount = Math.round((_level3.getBytesLoaded() /
_level3.getBytesTotal()) * 100);
            if(amount == 100){
                clearInterval(_root.preload_L3);
                _level3._root.KTZ__xml = _level0._root.KTZ__xml;
                _root.KTZ__LOAD();
            }
        }
        stop();
    actions for Symbol 548
        on (release) {
            _global.KTZ_USER = usr
            gotoAndPlay(2);
        }
    actions for frame 1
        stop();
    actions for frame 7
        stop();
Symbol Definition(s)
```

Interface: content manipulation

```
Scene 1
    actions for process
        onClipEvent (load) {
            this._visible = 0
        }
    actions for frame 1
        KTZ__LIMPIA = function () {
            for (var i in this) {
                if (i.indexOf("__KTZObject_") == 0 && i != "KTZ__paletaMC") {
                    // _root.removeMovieClip(_root[i]);
                    _root[i].matarse();
                }
            }
        };
```

```
KTZ__LIMPIA_UNO = function (objNombre) {
    trace(":.:::" + objNombre.substr(6));
    _root[objNombre.substr(6)].matarse();
};
KTZ__MOSTRAR = function () {
    if (!_root._noVer) {
        for (var i in KTZ_CLB_SOFrameActual.data) {
            if (i.indexOf("__ELEM") == 0) {
                var nombreOB = i.substring(6);
                ind = KTZ_CLB_SOFrameActual.data[i];
                mostrarKTZObject(ind.KTZtipo, ind.KTZ_color, ind.KTZnom,
ind.KTZcontenedorX, ind.KTZcontenedorY, ind.KTZcontenedorRota,
ind.KTZobjTOP, ind.KTZobjIZQ, ind.KTZobjDER, ind.KTZobjINF, ind, ind.KTZid,
ind.KTZobjTEXT);
            }
        }
    }
};
KTZ__MOSTRAR_UNO = function (objNombre) {
    if (!_root._noVer) {
        ind = KTZ_CLB_SOFrameActual.data[objNombre];
        mostrarKTZObject(ind.KTZtipo, ind.KTZ_color, ind.KTZnom,
ind.KTZcontenedorX, ind.KTZcontenedorY, ind.KTZcontenedorRota,
ind.KTZobjTOP, ind.KTZobjIZQ, ind.KTZobjDER, ind.KTZobjINF, ind, ind.KTZid,
ind.KTZobjTEXT);
    }
};
text = "__ELEM_EL RESTO";
KTZ__BORRA = function () {
    obj = _root[_root.itemsMSG.EDITA];
    delete obj.indice;
    trace(">>>>>>>>>>>>>>"+"__ELEM"+obj._<);
    delete KTZ_CLB_SOFrameActual.data["__ELEM"+obj._name];
    var e = false;
    for (var i in KTZ_CLB_SOFrameActual.data) {
        trace(KTZ_CLB_SOFrameActual.data[i] + ":" + i);
        if (i.indexOf("__ELEM") == 0 && KTZ_CLB_SOFrameActual.data[i] !=
null && KTZ_CLB_SOFrameActual.data[i] != undefined) {
            e = true;
        }
    }
    trace(">>>>>>>>>>>>>>"+e);
    if (!e) {
        trace(">>>>>>>>>>>>>>"+"FRAME"+_root._frameActual);
        delete ContentFrames["FRAME"+_root._frameActual];
        delete _global.usedFrames[_root._frameActual];
        if (_root._frameActual%5 == 0) {
            var ir = 2;
        } else {
            var ir = 1;
        }
    } else {
        var ir = 3;
    }

_root.KTZ__movieController.contenedor["f"+_root._frameActual].gotoAndStop(ir
);
    obj.matarse();
    _root.itemsMSG.EDITA = null;
};
KTZ__SAVE = function () {
    _level0._root.KTZ__SAVE();
};
KTZ__SEND = function () {
    _level0._root.KTZ__SEND();
};
KTZ__RELOAD = function () {
    _level0._root.KTZ__RELOAD();
```

```
        };
    actions for frame 1
        MovieClip.prototype.createKTZObject = function(tipo, posx, posy,
ancho, alto, col) {
            col = _root.KTZ__paletaMC.selectedColor.getRGB();
            if (ContentFrames["FRAME"+_root._frameActual] == undefined ||
ContentFrames["FRAME"+_root._frameActual] == null) {
                ContentFrames["FRAME"+_root._frameActual] = true;
                _global.usedFrames[_root._frameActual] = true;
            }
            if (KTZ_CLB_SOFrameActual.frame != _root._frameActual) {
                KTZ_CLB_connectToFrame(_root._frameActual, {obj:this,
fn:createKTZObject2, args:[tipo,posx,posy,ancho,alto,col]});
            } else {
                this.createKTZObject2(tipo,posx,posy,ancho,alto,col)
            }
        };
        MovieClip.prototype.createKTZObject2 = function(tipo, posx, posy,
ancho, alto, col) {
            var id = ++ContentFrames.__TKZOID;

//ContentFrames["FRAME"+_root._frameActual]["__ELEM"+"__KTZObject_"+id] =
new Object();
            KTZ_CLB_SOFrameActual.data["__ELEM"+"__KTZObject_"+id] = new
Object()

_root.KTZ__movieController.contenedor["f"+_root._frameActual].gotoAndStop(3)
;
            var ob = this.attachMovie("KTZ__BaseObject", "__KTZObject_"+id,
id+1000);
            _root.itemsMSG.addListener(ob);
            ob.tipo = tipo;
            ob._x = posx;
            ob._y = posy;
            ob.T_width = ancho;
            ob.T_height = alto;
            ob.T_color = col;
            ob.indice =
KTZ_CLB_SOFrameActual.data["__ELEM"+"__KTZObject_"+id]//ContentFrames["FRAME
"+_root._frameActual]["__ELEM"+"__KTZObject_"+id];
            ob.indice.KTZid = id;
            ob.indice.KTZnom = "__KTZObject_"+id;
            ob.indice.KTZtipo = tipo;
            ob.indice.KTZ_color = col;
            ob.indice.KTZcontenedorX = posx;
            ob.indice.KTZcontenedorY = posy;
            ob.indice.KTZcontenedorRota = 0;
            ob.indice.KTZobjTOP = 0;
            ob.indice.KTZobjIZQ = 0;
            ob.indice.KTZobjDER = ancho;
            ob.indice.KTZobjINF = alto;
            //ob.T__color.setRGB(col);
            ob._enabled = true;
            // esto no me gusta...
            ob.gotoAndPlay(2);
        };
        _root.createEmptyMovieClip("KTZ__checkCliks", 10);
        KTZ__checkCliks.onMouseDown = function() {
            this.hit = false;
            for (var j = _root.paletas.length-1; j>=0; j--) {
                if (_root.paletas[j].hitTest(_xmouse, _ymouse, false) &&
_root.paletas[j]._visible) {
                    this.hit = true;
                    j = -5;
                    break;
                }
            }
```

```
            //( _root.KTZ__paletaHerra.hitTest(_xmouse, _ymouse, false)) ?
this.hit = true : null;
          if (!this.hit && this.construyendo>0) {
              // chequear si NO es resaltador o flecha
              if (this.construyendo != 5 && this.construyendo != 6) {
                  this.lineaOn = true;
                  this.lp = _root.attachMovie("KTZ__ObjectLine",
"KTZ__LineaPunteada", 110000);
                      this.lp._x = _root._xmouse;
                      this.lp._y = _root._ymouse;
                      this.onEnterFrame = function() {
                          this.lp._xscale = (_root._xmouse-this.TMPiniX);
                          this.lp._yscale = (_root._ymouse-this.TMPiniY);
                      };
                      this.TMPiniX = _root._xmouse;
                      this.TMPiniY = _root._ymouse;
              } else {
                      _root.createKTZObject(this.construyendo, 0, 0, 10, 10,
0x666666);
              }
          }
          /*
              // esta es la parte de soltar el edit...pero no
              else if (!this.hit && this.construyendo == 0 &&
_root.itemsMSG.EDITA !=null) {
                      _root[_root.itemsMSG.EDITA].deseditar();
                      _root.itemsMSG.EDITA = null;
              }*/
      };                                                      `\
      KTZ__checkCliks.onMouseUp = function() {
          if (!this.hit) {
              if (this.construyendo>0) {
                  if (this.construyendo == 5 || this.construyendo == 6) {
                      this.construyendo = 0;
                      _root.KTZ__paletaHerra.onChangeNum();
                  } else {
                      this.lineaOn = false;
                      delete this.onEnterFrame;
                      removeMovieClip(_root.KTZ__LineaPunteada);
                      if (_root._xmouse>this.TMPiniX) {
                          var TR = _root._xmouse;
                          var TL = this.TMPiniX;
                      } else {
                          var TR = this.TMPiniX;
                          var TL = _root._xmouse;
                      }
                      if (_root._ymouse>this.TMPiniY) {
                          var TD = _root._ymouse;
                          var TU = this.TMPiniY;
                      } else .{
                          var TD = this.TMPiniY;
                          var TU = _root._ymouse;
                      }
                      var W = TR-TL;
                      var H = TD-TU;
                      (W<10) ? W=50 : null;
                      (H<10) ? H=50 : null;
                      _root.createKTZObject(this.construyendo, TL, TU, W, H,
0x666666);
                      if (this.construyendo == 4) {
                          this.construyendo = 0;
                          _root.KTZ__paletaHerra.onChangeNum();
                      }
                  }
              }
          } else {
              this.lineaOn = false;
              delete this.onEnterFrame;
```

```
                removeMovieClip(_root.KTZ__LineaPunteada);
            }
        };
        itemsMSG = new Object();
        ASBroadcaster.initialize(this.itemsMSG);
        //broadcastMessage("freeze");
        //addListener(this["lanza"+this.players[i]])
        _global.calculaDist = function(x1, y1, x2, y2) {
            return Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
        };
        _global._rtd = 180/Math.PI;
        ///
        MovieClip.prototype.mostrarKTZObject = function(tipo, col, nom, posx,
posy, rot, top, izq, der, inf, indice, id, tex) {
            var ob = this.attachMovie("KTZ__BaseObject", "__KTZObject_"+id,
id+1000);
            _root.itemsMSG.addListener(ob);
            ob.tipo = tipo;
            ob._x = posx;
            ob._y = posy;
            ob._rotation = rot;
            ob.reconstruye = 1;
            ob.R_top = top;
            ob.R_izq = izq;
            ob.R_der = der;
            ob.R_inf = inf;
            ob.T_color = col;
            ob.indice = indice;
            ob.R_text = tex;
            //ob.T__color.setRGB(col);
            (_root.KTZ__MODE == "TML") ? ob._enabled=true : null;
            ob.gotoAndPlay(2);
        };
        MovieClip.prototype.mostrarKTZGNotes = function(id, nom, tipo, color,
posx, posy, rota, TOP, IZQ, DER, INF, ALPHA, TEXT) {
            padre = this.createEmptyMovieClip("globalNotes", 1000005);
            padre.closed = true;
            padre.onEnterFrame = function() {
                _root.cerrarme(this);
            };
            ob = padre.attachMovie("KTZ__ObjectTipe"+tipo, "Objeto", 1);
            ob.R_x = posx;
            ob.R_y = posy;
            ob.R_width = der;
            ob.R_height = inf;
            ob.R_color = color;
            ob.R_text = text;
            ob.R_alpha = alpha;
            ob.R_global = true;
            padre.indice = KTZ_CLB_SOGlobalNotes.data
            /*padre.indice.KTZid = id;
            padre.indice.KTZnom = "__KTZGlobal_Note";
            padre.indice.KTZtipo = tipo;
            padre.indice.KTZ_color = col;
            padre.indice.KTZcontenedorX = posx;
            padre.indice.KTZcontenedorY = posy;
            padre.indice.KTZcontenedorRota = 0;
            padre.indice.KTZobjTOP = 0;
            padre.indice.KTZobjIZQ = 0;
            padre.indice.KTZobjDER = der;
            padre.indice.KTZobjINF = inf;
            padre.indice.KTZobjALPHA = alpha;*/
            padre.indice.KTZobjTEXT = text;
        };
        MovieClip.prototype.setKTZGLevel2 = function(id, posx, posy, scale) {
            _level2._x = posx;
            _level2._y = posy;
            _level2._xscale = _level2._yscale=scale;
```

```
            _root.KTZ__paletaControl.inicio();
            padre = this.createEmptyMovieClip("levelHolder", 1000006);
            padre.indice =
ContentFrames["globalObjects"]["__ELEM__KTZGlobal_level2"];
            padre.indice.KTZid = id;
            padre.indice.KTZnom = "__KTZGlobal_level2";
            padre.indice.KTZcontenedorX = posx;
            padre.indice.KTZcontenedorY = posy;
            padre.indice.KTZobjSCALE = scale;
        };
    actions for frame 1
        cerrarme = function (pal) {
            if (pal.closed) {
                _root.KTZ__paletaHerra.closePal(pal);
                pal.ready = true;
            } else {
                _root.KTZ__paletaHerra.openPal(pal);
                pal.ready = true;
            }
            (pal.ready) ? delete pal.onEnterFrame : null;
        };
        _root.paletitas = new Array();
        _root.paletitas.push({movie:"KTZ__paletaColor", name:"KTZ__paletaMC",
posX:300, posY:300, closed:true});
        _root.paletitas.push({movie:"KTZ__paletaOpciones",
name:"KTZ__paletaOpciones", posX:150, posY:300, closed:true});
        _root.paletitas.push({movie:"KTZ__paletaControl",
name:"KTZ__paletaControl", posX:75, posY:400, closed:true});
        _root.paletitas.push({movie:"KTZ__reloj", name:"KTZ__reloj", posX:664,
posY:379, closed:false});
        _root.paletitas.push({movie:"KTZ__paletaHerra",
name:"KTZ__paletaHerra", posX:1.9, posY:3, closed:null});
        function attachPaletas() {
            _root.paletas = new Array();
            depth = 1000009;
            for (var i = _root.paletitas.length; i>=0; i--) {
                pal = _root.attachMovie(_root.paletitas[i].movie,
_root.paletitas[i].name, ++depth);
                pal._x = _root.paletitas[i].posx;
                pal._y = _root.paletitas[i].posy;
                _root.paletas.push(pal);
                _global.paletteMSG.addListener(pal);
                _root.highest = pal;
                if (_root.paletitas[i].name != "KTZ__paletaHerra") {
                    pal.closed = _root.paletitas[i].closed;
                    pal.onEnterFrame = function () {
                        _root.cerrarme(this);
                    };
                }
            }
            _root.attachMovie("KTZ__borrador", "KTZ__borradorMC", 100+depth);
            KTZ__borradorMC._x = -100;
            KTZ__borradorMC._y = -100;
            ktz__reloj.reloj.play();
            delete _root.paletitas;
        }
        _root.controllerExists = false;
        function attachController() {
            if (KTZ__MODE == "TML") {
                attachear = {movie:"KTZ__paletaMovieController",
name:"KTZ__movieController", posX:399, posY:490};
                removeMovieClip(_root.KTZ__reloj);
                _root.controllerExists = true;
                KTZ_setEnable();
            } else if (KTZ__MODE == "KEY") {
                attachear = {movie:"KTZ__paletaKeyController",
name:"KTZ__movieController", posX:1.9, posY:3};
                _root.controllerExists = true;
```

```
        }
        maxDepth = _root.highest.getDepth()+1;
        pal = _root.attachMovie(attachear.movie, attachear.name, maxDepth);
        pal._x = 399;
        pal._y = 490;
        _root.paletas.push(pal);
        _root.highest = pal;
        delete pal;
        delete attachear;
    }
    stop();
actions for frame 1
    keyListener = new Object();
    keyListener.onKeyDown = function() {
        trace(Selection.getFocus());
        if (!_root.editText && Selection.getFocus()
!="_level4.chatBox.txtBox") {
            if (Key.isDown(Key.DELETEKEY) || Key.isDown(Key.BACKSPACE)) {
                KTZ__BORRA();
            }
            if (Key.isDown(86)) {
                // V --> select
                _root.KTZ__checkCliks.construyendo = 0;
                _root.KTZ__paletaHerra.onChangeNum();
            }
            if (Key.isDown(84)) {
                // T --> text
                _root.KTZ__checkCliks.construyendo = 4;
                _root.KTZ__paletaHerra.onChangeNum();
            }
            if (Key.isDown(70)) {
                // F --> filledObject
                _root.KTZ__checkCliks.construyendo = 2;
                _root.KTZ__paletaHerra.onChangeNum();
            }
            if (Key.isDown(16)) {
                // SHIFT para el movimiento en x e y del level
                _root.shiftDown = true;
                KTZ_setEnable();
            }
        }
    };
    keyListener.onKeyUp = function() {
        _root.shiftDown = false;
    };
    Key.addListener(keyListener);
Symbol Definition(s)
```